



## Cual collection usar

### Transcripción

[00:00] Hola, hola. Vamos a iniciar nuestra clase 17. En nuestra clase 17 vamos a ver cual collection usar y va a ser nuestra última clase de otros tipos de sets y create. Para eso vamos a duplicar aquí, vamos a colocar aquí clase 17 y vamos a ver lo siguiente. En resumen, nosotros hemos visto aquí la clase collection y hemos visto nuestras interfaces.

[00:29] Serían set y list. En set hemos visto hashSet y linked hashSet. En list hemos visto linkedList y arrayList. Una vez hecho esto, tenemos que saber en qué momento usarlo o cuándo usarlo. Recuerdan que en nuestra clase, creo que fue en nuestra clase 5, vamos a ver aquí. No fue en la 6, no fue en la 7, vamos a ver aquí en la 3. Fue en la 8.

[01:02] ¿Recuerdan que en la 8 estuvimos viendo tiempos para ver cuál era mejor o cuál usar de acuerdo a la situación? En la misma cosa lo que vamos a hacer prácticamente lo que es ahora, saber decidir. ¿Por qué? Porque los iterators es una forma en que nosotros podemos recorrer una lista. Esa es una cosa, podemos utilizar también el forEach de lambda.

[01:31] Podemos utilizar también otro forEach. Vamos a ver aquí. Podemos utilizar también otro forEach, de for. Podemos utilizar también otro tipo de for para recorrer la posición. Tenemos un montón de formas de poder hacerlo. Tenemos varios tipos de formas en realidad.

[01:57] Entonces, pensando en esa forma, si nosotros no sabemos mucho cómo usarlo o cuál va a ser el objetivo, siempre es bueno ver todo lo que es siempre

documentación. Entonces no sabemos si vamos a trabajar con set, con list, entonces nuestro atributo va a ser un collection y cualquier situación, lo cambiamos.

[02:20] Ahora, si ya sabíamos, si es set, tenemos que también utilizar nuestro equals y hashCode para poder hacer las comparaciones por el hash, qué es lo que el HashSet y el linked HashSet piden. Entonces existen esos pequeños detalles que de acuerdo a nuestras necesidades van a surgir por qué usar uno, por qué usar el otro.

[02:48] En la mayoría de las situaciones va a ser también un poco de performance, porque por ejemplo queremos adicionar una lista, un valor a una lista o queremos modificar una lista, tal vez más rápido sea el HashSet. Queremos remover, tal vez los dos sean iguales. Queremos la posición tal, tal vez sea mejor el list. No nos interesa la posición, puede ser el HashSet.

[03:16] Queremos recorrer una lista con un while, podemos utilizar el iterator. Queremos recorrerlo con lambda, podemos utilizar el forEach de lambda. Queremos utilizar filter, podemos utilizar el forEach del mismo modo. Podemos convertir nuestra lista en string y utilizamos el filter también.

[03:35] Tenemos muchas opciones. Básicamente sería saber en qué momento usarlo y cuándo usarlo. Estuvimos viendo en los ejemplos, por qué usar uno, por qué usar el otro. También tenemos nuestra clase 8, que les va a servir bastante para ver los tipos, para probar. Quiero probar con tanta cantidad de registros, para ver cuál es mejor, cuál el peor, cómo funciona el add, el get o el remove y vamos a comparar.

[04:05] Y ahí con eso ya podemos saber cuál nos puede ayudar de acuerdo a la situación que estamos necesitando. Ese es un punto muy importante. Entonces espero que haya sido claro. Cualquier cosa siempre es bueno abrir la documentación, ver todo lo que es documentación, porque todo está aquí. Y cualquier cosa que necesitemos, podemos encontrarlo aquí mismo también.

[04:31] Entonces esto sería toda nuestra clase 17, que básicamente hicimos un resumen general, y nos vemos en nuestra siguiente clase. Muchas gracias.